Eternal Blue system hack notes:
7NOV21

###
code segments are mapped with read-only access permissions, and are verified during the secure boot process. This means that once TrustZone's code is loaded into memory, it theoretically cannot (and should not) be subject to any change.

{trust zone, cortex A
https://en.m.wikipedia.org/wiki/ARM_architecture#Security_extensions}

“the vulnerability allows the attacker to cause the TrustZone kernel to write a zero DWORD to any address in the TrustZone kernel's virtual address space.”

it means it can override any software security feature

https://www.elitepvpers.com/forum/black-desert/4550027-black-desert-dword-lite-hack-v1-0-08-11-2018-a.html

bc it has direct access to core bus;

“remote code execution”
https://www.exploit-db.com/exploits/42315

due to a “flaw” in the chip architecture ; aka useless fucking junk

###
====
(This is what it’s doing; how its hiding itself.)
theoretically, if the bios wasnt an issue (im not saying it isnt), and you had a clean host; you could write a custom driver, and wipe the drive and reuse the hardware—
====
The kernel expects to be dealing with devices that implement 512-byte sectors. If your device uses a different size, the kernel adapts and avoids generating I/O requests that the hardware cannot handle. It

Block drivers, must use a set of registration interfaces to make their devices available to the kernel.

registration interface is first thing i would counter-target.

The function for this task isregister_blkdev (which is declared in <linux/fs.h>):

 The arguments are the major number that your device will be using and the associated name (which the kernel will display in /proc/devices).

block driver arguments re /proc/devices is where the bad code is hidden. easy enough to fix

A block driver might respond to an open call by spinning up the device, locking the door (for removable media), etc. If you lock media into the device, you should certainly unlock it in the releasemethod
###
Also see:
{https://unix.stackexchange.com/questions/39113/mount-ntfs-image-file-created-using-partimage} (Loop device, virtual partition; ntfs disk image, mounted on linux)

—

{https://www.oreilly.com/library/view/linux-device-drivers/0596005903/ch16.html} (O Reilly Chap. 16, Block Drivers, linux)
—

{https://bits-please.blogspot.com/2015/08/full-trustzone-exploit-for-msm8974.html?m=1} (Kernel memory exploits)